|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| To Tame The Void  Design Document |  |
|  |  |

# Table of contents

*I. Table of contents* 2

II. Introduction 6

1. General Description 6

2. The Galaxy Map 6

3. Wining Conditions 6

4. Philosophy 6

III. Game Modules 7

IV. Empires & Relations 9

1. The trust scale 9

2. Diplomacy 10

V. In-Map Units 11

1. Imperial Headquarters (IH) 11

a. The building queue 11

b. HQ Options 11

2. Task Forces 12

a. Manipulation 12

b. Movement 13

VI. Resources and Production 15

1. Stars Systems 15

a. Model 15

b. Description 15

2. Strategic resources 15

3. Outposts 16

a. General definitions 16

b. Effects 17

c. Upgrades 18

4. The Reserve 18

a. Empire-Level Policies 18

b. System Actions/Projects 18

c. Units 19

5. Growth Policies 19

VII. Expansion and Control 21

1. Design Criteria 21

2. Basic Combat - Minimum Viable Game 21

a. Ship Attributes 21

b. Combat 22

c. Available Ships 22

3. Advanced Ships - Long Term Development 22

a. Ship Attributes 22

b. Weapon Attributes 23

c. Factory Attributes 23

d. Special Attributes 24

e. Example Ships 24

4. Space Combat 25

a. Engaging the enemy 25

b. The Combat Turn 25

c. Choosing Targets 25

d. Applying damage 26

e. Choosing strategies 28

f. Retreating 29

5. Other 29

6. Conquest 29

a. Troops 29

b. Invasion 29

VIII. Intelligence and Influence 30

a. General Description 30

b. In enemy territory 30

c. Units 30

IX. Technology (TODO) 31

a. Space Warfare 31

b. Xeno-Domination 31

c. Development 31

d. Exploration 31

X. Story (TODO) 31

1. Race 31

a. Concepts 31

b. Preliminary Ideas 31

2. Storyline 31

XI. Roadmap 31

XII. Turn & Event Processing 31

1. State 31

a. Concepts 31

b. Procedure 31

2. Turns 31

a. Possible Order 31

b. Step details 31

3. Actions & Events 31

a. Context 31

b. Proposals 31

c. Event list 31

XIII. Interfaces 31

1. The star control 31

a. Overview 31

b. Actions 31

2. The fleet control 31

3. The IHQ control 31

XIV. AI 31

1. Pre-requirements 31

2. Production Assignment 31

a. Allocation amount 31

b. Overshooting and undershooting 31

3. Production Choices 31

4. Strategic objectives 31

5. Object Classifications 31

6. Tactical Objectives 31

7. Accumulators (code) 31

XV. Sandbox 31

1. System Developments 31

2. Spending 31

a. Localized developments 31

b. Global 31

3. Races 31

a. Modifiers 31

4. Empire Attributes 31

a. Fleet design 31

5. Space Combat 31

# Introduction

## General Description

To Tame the Void is a galaxy conquest game. The game starts with two or more players, each one representing a galactic empire and starting on its own home planet. A player can discover technologies and create task forces to travel to other stars in the galaxy, discovering, colonizing and eventually finding other players and destroying or ruling them.

## The Galaxy Map

The galaxy map is the main interface of the game, showing a graph of star systems connected through starlanes. Starlanes are the only way of traveling between stars and are strategically important, since they create chokepoints so not every star need a defending task force or risks being attacked at any time. The density of starlanes could be configurable at game start.

The galaxy map can be zoomed in or out to concentrate on parts of the map. There are two levels of zooming. The kind of interaction the user can get at these levels is actually identical, but the amount of information, and how it is displayed, diminished if we zoom out.

The most important interfaces should be overlaid directly over the map, avoiding the need to constantly go into separate screens. This goes for configuring and sending task forces, checking star characteristics, building ships, etc. More complex and less frequent actions like project building, ship design, tech selection and spending should be full screen.

## Wining Conditions

The main objective of any given player is to become the ruler of the galaxy. In practice, this can be attained in the following ways:

* Annihilation: All adversary empires are destroyed.
* Control: All adversary empires are under your influence to a point that they are considered puppets.
* Expansion: Your Empire controls, either directly or through puppet empires, 2-3 of the star systems in the galaxy, making any resistance futile.
* Brotherhood: all remaining players are extremely friendly and long-term peace has been achieved.

## Philosophy

The main experience intended by the game is on of total immersion on the global map. Interfaces and features are designed so the player does not have to switch from the macro level (empire / galaxy) to higher levels of detail. Some common feature of 4x games are avoided completely (per-system buildings and construction queues) and other greatly streamlined (technology, diplomacy, espionage) as to avoid spending a lot of time outside the main screen.

Simultaneously, the main purpose of the game is to feel BIG. The game should be fast-paced and avoid long turn times while simultaneously allowing fleets of thousands of units to confront each other in battle.

# Game Modules

The idea of the game is that it should be highly modular. This also allows for incremental implementation or future moding. The problem with this approach is that any interesting definition for the different modules will cause interdependencies. For instance, if a certain model for economic production is used, resulting on several types of resources, then any other part of the game that requires resources needs to understand it. A practical solution is to make most modules completely independent from each other, while fixing the interaction between others. Potential modules can include:

|  |  |  |  |
| --- | --- | --- | --- |
| Module | Priority | Type | Description |
| Base economic model | 1 | Economy | Basic model for growth were a single resource is created. |
| Base unit model | 1 | Unit | Basic system for moving units between stars. Each unit has a type, can be attached or detached from a star and can be instructed to travel between stars. A unit (a task force for instance) is composed of stacks of sub-units, with a counter for the number of them. |
| Basic star model | 1 | Map | Basic map model. Each star can hold a number of units, sorting them by owner and type. Units can then require a slot in orbit and information about the presence of others in the same star. Stars can have a single outpost. |
| Basic construction model | 1 | Production | A simple production system were an "Imperial Command" can be built on a certain star. All units are produced at such centers, to avoid micro-management of hundreds of stars. |
| Basic task force model | 1 | Unit | A system were a predefined set of ship models is available. Ships types are limited to a small set of attributes (damage, aim, dodge, etc.). The task force has a single "military might" status. |
| Basic combat model | 1 | Combat | Units cause damage in a fixed number of rounds before the fight is considered over (can continue next turn). No simulation is shown, just the result. |
| Basic invasion module | 1 | Invasion | A single unit, "troop carrier" allows invasion of outposts. A random set of infrastructure is destroyed and another conquered per carrier. If all the remaining infrastructure is conquered, the outpost switches owner. |
| Basic Technology | 2 | Tech | Each empire contains a specific set of multipliers, starting at 1.0 and incrementing slowly (5-10%). Production can be spend in tech in order to increment these values. |
| Unlockable components | 3 | Tech | Same as basic tech, but attributes are now grouped in a few categories, and a tech level assigned to each. Research happens on categories (as Moo1) and also, special unlocks occur at specific levels. |
| Influence Module | 2 | Influence | Allows an imperial command to include a "shadow academy", able to build a new set of units: infiltrators, spies, smugglers, etc. |
| Animated combat | 2 | Combat | Combat is animated as a 1D battle, with ships only advancing from either side, until merging in the middle. Allows players to better determine the composition and efficiency of task forces. |
| Real time combat | 3 | Combat | Each stack of units is allowed basic behavior and orders, like advancing, holding ground, retreating, etc. |
| Animated Invasion | 2 | Invasion |  |

# Empires & Relations

## The trust scale

The basic measurement of the relationship between two empires is how much they trust each other. If an enemy empire is trusted; trade and other mutual benefits are exploited more, war is less likely to happen, frontiers are more open, etc.

#### Mechanics

For two empires A and B, A chooses how much it trusts B, B chooses how much it trusts A and the level of mutual trust is determined by the lesser quantity. The minimal common level of trust is then used to determine how open is the frontier between these empires and hence how easy is for influence units to enter or infiltrate the opposite empire. It is important hence to choose you allies carefully, since trust is a double edged sword. On one hand it can bring great rewards, on the other it can bring great danger.

The trust scale changes when either empire desires to change its attitude to the other, but in practice such a change in policy cannot be immediately applied across whole empires, and becomes effective only gradually. Trust being measured between 0 (no trust) and 1 (total trust), any change in policy can only shift at a rate of 0.1 per turn, so it could take to 10 turns for a shift in trust to fully affect influencers.

#### Secondary effects

There are certain thresholds in the trust scale that unlock new options for both empires. A certain level of trust becomes tempting given the many additional benefits. These benefits, in order, are:

* Cease Fire: fleets will not automatically engage and can coexist peacefully in unclaimed systems. Fleets that find themselves in foreign systems will retreat automatically.
* Open Trade: trade is permitted between the empires, making traders safe to travel into the foreign empire and increasing their effectiveness.
* Cultural Cooperation: scholars can go into the foreign empire in order to learn about their culture and technology.
* Sanctuary: Foreign units are considered friendly and will be given assistance (including repairs) in frontier outposts. Friendly units can stay in orbit for long periods.
  + If a third fleet attacks the system, the sanctuary is enforced. A third empire is as such unable to attack an adversary in friendly territory without attacking both fleets.
  + Two friendly fleets that are not friends between each other can't engage while in their friend's system.
* Special Resource Sharing: traders from an empire get access to important strategic resources owned by the friendly empire.
* Open Borders: friendly empires can travel between stars belonging to this empire.

## Diplomacy

Diplomacy, while normally part of most 4x games, is not a desirable game concept. These are some personal arguments against it:

* It feels separate to the main flow of the game.
  + It can occur at any time or not at all.
  + It has little to no relation to other mechanics in the game. Even trying to tie it down to technological advance (which is by definition pretty broad) results in weird tradeoff like making a space-faring civilization have to discover “peace treaty” and basic stuff like that.
* It’s not particularly engaging as a feature.
* If it needs to be really exploited, it’s a time consuming and tedious endeavor.
* It requires significant additional AI. Some games allow pretty much any kind of deal to occur, but in practice many are always denied.
* It is slow paced.
* It’s a kind of micro-management when many adversaries are present.

If any kind of diplomacy is eventually added, it should better be reduced to a single screen related to the situational report, were basic actions like opening borders, declaring war, offer peace or establish basic trade.

# In-Map Units

## Imperial Headquarters (IH)

To limit the micromanagement in the game, and to create an additional strategic consideration for the game, units are created in a few specific locations only. An imperial headquarter is an expensive system project that adds, after completion, provide with 4 production slots. Each slot can be dedicated to the creation of a single type of unit at the time, and the unit appears at that location. The cost of the unit is the charged to the reserve. Each slot includes:

* A type, which in turn defines the units that can be produced in it. Available types depend on specific modules, but examples include:
  + Orbital Shipyard: produces ships for task forces.
  + Marine Barracks: produces ground troops for invasion and defense.
  + Shadow Academy: produces spies and other influencers.
* A level, which determines the total number of resources that the slot can churn per turn. The slot can be upgraded to increase production. Of course, a higher level command headquarter is a juicy target.

IHs are marked with their own distinctive icon on the galaxy map can be found orbiting stars along task forces, they also have their own specific interfaces.

### The building queue

The original idea was for each slot to be able to produce a unit of the specific type. Upgrades and construction speed would be handled at the slot type and multiple slots built independently. This ended up being not only harder to implement, but also too complex in both implementation and interface (sorting slots, empty slots, separate upgrades, etc.).

An alternative is to decouple available types and HQ level from the slots themselves. So, a single HQ can have several types of independent upgrades:

* Modules: increasing the available units on every slot.
* Slots: making a new slot available, allowing production in parallel.
* Level: increases total output from the HQ, to be distributed among slots.

These upgrade options, being fewer, can be better integrated with the standard project UI in each star.

### HQ Options

#### Relocation

Upon construction, the unit starts with the assigned route as order. The interface to create the route is the same as the one to determine a fleet's route. More details on how paths are created are seen in the fleet's movement section.

#### Output

A button allows to toggle the current output of the HQ. By default each HQ builds unit at the maximum possible speed, as long as there are enough resources to do so. This option allows to halve or halt construction in order to save resources. HQ Maintenance though still exists.

## Task Forces

While units are produced independently, it is desirable to group them in order to manipulate them more easily on the galaxy map. Different task forces are created depending on the type of unit, depending on the type of lost that produces them at the IH. This allows creation of separate behaviors for a task force depending on its type. Upon being produced at an IH, a unit goes into the first task force in orbit with a matching type; else a new task force is created for such purpose.

Units of the same type are stacked together into a single count, further simplifying handling huge numbers of units[[1]](#footnote-1). In order to reduce the number of task forces required to handle units, as well as too much clutter in the map, every time two task forces of the same type and owner coincide orbiting the same star, they are merged together. This is sometimes referred as the "default" task force of a system.

Task forces are only visible as a single icon on the main map, no matter how many ships are contained inside a particular task force. This avoids clutter and simplifies issuing commands. On the other hand, the same ability to stack multiple ships together may hide important information from the player, who needs to be able to determine the balances of power. For this reason, it might be convenient to divide task force icons into three or so distinctive icons, with at least a 20x increase in total task force base cost between them.

### Manipulation

Task forces can be either orbiting a star or in transit between stars. A task force in transit cannot be manipulated on any way, but once back in orbit a subset of units (by default all of them) can be selected in order to do one of several possible manipulations.

#### Selection

Clicking on a task force icon opens a special dialog, detailing each stack in it by showing the unit icon and its count. Selection is done by clicking on these icons. Some considerations:

* Initially, all stacks are fully selected the number representing the total number of units of each type.
* Clicking on the icon changes the number of selected units, decrementing them in case of a left-click and incrementing them in case of a right-click.
  + One more click past the maximum value cycles to 0 and vice-versa. The full number is divided in 12 increments (about 8.3% per click), rounding each incremental value.
  + For small numbers of units, a minimum of 1 unit is always selected.
  + Holding the Ctrl key while clicking toggles 3 increments on each click.
* A special button is also available to toggle between all or none of the units across all stacks.

#### Options

Once a selection is made, special buttons on the dialog allow the following actions:

* Movement to a different system, by clicking on it.
* Creating a named task force, which is never merged with others by default, it remains a separate entity. Useful to avoid reselecting tasks forces who temporarily coincide on a star system.
* Scrap them, returning part of their construction cost to the reserve.
* Moving them into another task force in the same system by left-clicking on its icon. This allows disbanding a named fleet or moving units between task forces.

Task forces can be sent from one star to another by clicking on the task force icon and then on the desired destination, which should of course be connected to the current star by a starlane. As a convenient shortcut, clicking on a destination can be done while ships are in the staging area. If this is done, it will automatically create a new task force and sends it to the selected destination, but upon arrival the task force will automatically disband. These temporary task forces are useful to avoid too much micromanagement and can be used all the time without ever creating new task forces. The ability to define task forces can be convenient nevertheless to avoid having to reselect stacks of ships when more than one task force reaches the same star or other ships are already there.

### Movement

#### Hyperspace travel

The basic system of space travel involves the use of a hyperspace gate to allow movement through hyperspace along a starline/fracture. A hyperspace gate is opened by the engine and the shield guarantees that the task force can remain cohesive under the strong ripping forces of the hyperspace. The shield can encompass the same ship where it’s generated or nearby ships of the same task force. Task forces using hyperspace shield take several turns in transit, can be reached for orders with sufficiently high technology and can be detected from the other end. Ships can retreat from combat by opening a hyperspace gate and crossing it. Hyperspace launchers can accelerate ships through hyperspace if built on the origin. Hyperspace dampeners can reduce it at destination by polluting the starlane with hyperspace noise. The ability to open a hyperspace gate and the hyperspace shield are different ship components, and only a single gate is needed for all ships. Static gates

#### Jump Drives

A non-standard alternative of travel, it involves a jump drive that takes time to charge, slowly building up a time-space fracture. When the buildup is complete, all energy is released at once and space-time is ripped as if punched by a needle, teleporting the task force instantly to destination. Jump drives have the advantages of allowing cancelation of the buildup at any time and not being detectable in advance, exceling at surprise attacks. Disadvantages include not being able to retreat from combat (as a new jump would require a new slow buildup) and the fact that the energy required for the jump depends on both the mass of the task force and the distance, limiting the size of the moving task force further. Also, there are fewer technology improvements than in hyperspace technology. Gravitational stabilizers can be used as defenses in the destination system, but instead of slowing down jumps these devices make them more unstable. A random fraction of incoming ships could be lost in transit, never to be found again.

#### Other issues

The game doesn’t automatically calculate routes between stars. This design conclusion is due to the diversity of conditions and assumption for this problem. While pathfinding using starlanes is relatively simple, the intended path will depend on enemy presence, patrolling, strategy, etc. To simplify this problem while allowing total control to the player, a series of stars can be chosen instead of a single destination. A click on a star contiguous to the last star in the path (or the starting star in case of being in orbit) will add it to the path, while a right-click on any star in the path will delete all stars from that one onwards. The path always contains all the stars that a particular task force will move to.

# Resources and Production

## Stars Systems

### Model

The aim of the game is to consider a star system as the basic unit, usually referred to just as a star. A star system is nevertheless a complicated concept, since the characteristics of the star itself may be very different (red giant, binary star, etc.) and the same applies to the planets, asteroids and other celestial bodies that may be present in the system. To avoid considering all possible materials, minerals, energy sources, ecosystems and so on, we simplify the model of a star to a set of three attributes:

#### starSize

This attribute measures the sheer size of the system. A standard system will have a size of 0.5, the maximum possible system (representing many planets for example) has a size of 1.0. A size of 0.0 indicates the absence of habitable locations.

#### starConditions

This attribute indicates the inherent difficulty of colonization for the system. It includes all imaginable aspects ranging; presence of atmospheres, corrosive elements, storms, hostile native life, gravity, temperature, etc. Conditions range from 0.0 (terrible conditions) to 1.0 (very auspicious).

#### starResources

This attribute measures the richness of the system in terms of energy, minerals, life forms, etc. Even if conditions for colonization are tough, a rich system may be an interesting investment due to its resources. Resources go from 0.0 (only dirt) to 1.0 (unimaginably rich).

### Description

To avoid making stars too generic, each one should be accompanied by a longer description and iconography, showing for example a representative landscape that varies with the presence of a outpost. Examples could include a starbase made of asteroids, a desert, a submarine dome, a jungle, etc. This iconography is shown in secondary screens that relate to the star.

## Strategic resources

Strategic resources are rare findings that go beyond the scope of the normal standard resources that go into a outpost’s production. Strategic resources are extremely rare and influential and as such provide a galaxy-wide competitive advantage to an empire. A huge variety of resources and effects could be invented and each one should be unique galaxy-wide. Some arbitrary examples:

* Gaia: incredibly abundant and fertile ecologies are able to produce an extraordinary amount of sustenance, making this star a global source of food and thus improving conditions by 5%.
* Semi-Sentient telepathic crystals: native life that can be tapped to increase any efforts of espionage or influence other empires by 10%.
* Unobtanium deposits: Incredibly hard allow that can’t be obtained anywhere else in the galaxy, increases the toughness of all our ships by 10%.
* Spice Melange: drug that can heighten intelligence, improving technological efforts by 8%.

The frequency of strategic resources should be configurable when a galaxy is created but they should always be rare. The idea is to make some stars extremely desirable and thus produce conflict for their control. There should never be more than a single strategic resource in any star and the bonus should not be too large, normally between 5% and 10%.

## Outposts

An outpost ship needs to be built to establish a permanent base of operation in a system. An outpost ship is basically a big, expensive and slow moving ship, capable of bringing a significant amount of population and infrastructure to a new star system. Once an outpost ship is deployed at a particular destination, many of its components are cannibalized and repurposed to serve the needs of the outpost, so it can’t be reused to move to another destination. Once established, outposts usually provide a base of operation to provide for logistics, supplies, storage, commerce, repairs, long-range scanners and many other basic functions. Once established, outposts start to generate both production and costs, with a total output that affects the empire as a whole either positively or negatively.

### General definitions

Note: the formulas below depend on the base attributes of the star the outpost is on. One objective is that better stars influence better outposts, but not linearly, since the best outpost could then be worth a thousand bad outposts. Hence, all of a star’s attributes get a handicap of +0.75 before their inclusion on the following formulas. The variables P and M are constants determined by the current technological level and strategic resources, starting with 1.0 and increasing in small percentages.

#### Infrastructure

Infrastructure measures the size of the outpost, including factors like the number of colonists, habitation, factories, etc.

#### Production

Production is the total output of the outpost to the empire.

#### Maintenance

Total cost to maintain the infrastructure of a outpost

#### Conditions

Conditions start at the star's base conditions but deteriorate the more space is being used.

#### Construction costs

Cost of generating a new unit of infrastructure

### Effects

The formulas described above work together to produce a simple economic model. Infrastructure gets built first on the most auspicious locations for colonization, then on the remaining locations (were conditions are worse development and maintenance costs increase). In practice, a point can be reached were the production is maximized. In this case, the best possible outpost produces around 12.6 times the production of the average outpost. Star size, conditions and resources behave similarly on all cases and can be directly compared to get an intuitive idea of the promise of a particular star for colonization.

There are, nevertheless, some practical differences between the attributes. Better conditions will mean cheaper initial development. A bigger system means more stable development, since conditions will deteriorate much slower. A richer system means more output without needing to develop a lot of infrastructure, but such a system will be easier to invade.

Note that outposts are supposed to avoid all micromanagement. There is no “buildings” or choices of any kind associated with a single star. The idea is to keep the strategy at a global level, keeping stars only as territory that produces a sort of income.

#### Maximum Infrastructure

Given the formulas above, a maximum infrastructure is guaranteed to exist, and it can be calculated using derivatives to be:

#### Maximum Production

Given the maximum in infrastructure mentioned, the corresponding estimated maximum in production increases linearly with size, but varies depending on conditions and resources. In fact, very poor conditions or resources (or both) can make the cost of maintaining infrastructure greater than the expected production, making a system a net cost instead of a gain. Of course, a player can decide to pay such a cost for strategic reasons or to wait till technological advances make it a desirable option. The general behavior is shown in this graph:



Note: if the differences between systems are deemed too steep, it might be convenient to calculate the final production number as the square root (or other power function, i.e. 0.75) of the current production, making the differences more linear (3.54 times between the best possible system and the average one). Master of Orion, for comparison, has a total difference of 9x increase in production from ultra-poor to ultra-rich planets and about a 5-6x difference from planet size.

Note: it might also be useful to generate attributes based on a normal, rather than a uniform distribution.

### Upgrades

Depending on the modules being used, the technological constants M and P might vary over time. There are two possible strategies to follow here:

* **Instantaneous Upgrade:** Variable changes are applied inmediately on a new turn, when growth and production is calculated.
* **Invested Upgrade:** Local versions of the last values of M and P are stored, and before creating any new infrastructure money can be spent to upgrade existing infrastructure. The upgrade cost is then proportional to the increase on efficiency obtained and the expense determines a new average value for M and P, which can reach up to the maximum value set by tech level.

## The Reserve

Each empire has a global production reserve, were production from all outposts go. From this reserve are then taken all expenses, such that the reserve can either grow or shrink over time. The only exception is infrastructure growth, which is considered locally before production is moved to the reserve.

The reserve has to always be a positive number, as overspending or any other cause for lack of resources can lead to empire-wide problems. Sources of expenses can be basically classified into three categories:

### Empire-Level Policies

A diversity of game modules can post expenses to the reserve each turn. The expenses can be due to maintenance, growth, etc. If the reserve is empty and cannot cover these expenses, the module gets its postings rejected and handles the corresponding consequences.

### System Actions/Projects

While some modules work across all the empire without being tied to a specific location in the galaxy map, others need to be placed. In a few occasions an action can be taken at the system level that can also post a cost to the reserve (it there are sufficient funds). These actions can involve:

* System Improvements that shape the layout of the galaxy, for example:
  + Hyperspace Gates
  + Hyperspace Tunnels
  + Hyperspace Scanners
  + Hyperspace Dampeners
  + Jump Gates
  + Etc.
* Offensive Actions in systems claimed by an enemy empire, for example:
  + Sabotage
  + Bombing
  + Invasions
  + Etc.

Note that some of these options may require the presence of specific units in the star system and might not involve a separate expense that the normal unit maintenance.

### Units

Another source of expenses corresponds to game units. Game units do not work at the empire level, nor are they fixed to a specific star system. They are separate entities that not only can be produced in quantities, but can also move around the galaxy map. Units may incur in a certain cost per turn, can enable system actions with associated costs or can even increase production (i.e. traders).

## Growth Policies

Even if infrastructure and production are independent on each system, once on the reserve it makes sense for each empire to have a common policy to determine how many resources are spend on growth and how many on other things. Avoiding too much complexity, the central growth policy depends on 3 factors:

* P: The percentage of total empire production available for spending in further growth.
* T: A maximum time for any new infrastructure built to pay for itself.
* E: Permission to spend external production from the reserve to pay for new infrastructure.

When a turn is ended, the new production for each outpost is calculated (and its total). Then the growth policy is applied in three stages:

* First, each outpost spends its own production in the specified percentage P and builds new infrastructure. The limit T is respected, potentially diminishing the expense. This pass guarantees that all outposts have an equal opportunity to grow.
* Second, since some of the total available production might still be available, outposts get another chance to spend it, limited by their own local resources. Only outposts that have not hit the specified limits get a chance, in random order, until everyone had a chance or available production runs out.
* Third, if there is still available production (maybe because a lot of outposts have hit the limit but produce a lot) and spending of external resources is allowed, the second phase is repeated. This time the cost of new infrastructure is doubled (due to having to ship resources to the system), but the outpost expense limit increases from its own local production to twice that much.

# Expansion and Control

Expansion and control are achieved by building massive fleets, with hundreds if not thousands of ships. These fleets move between systems and clash between each other in order to gain or defend territory and the resources that come with it.

## Design Criteria

Handling massive fleets is no easy task. Traditionally, it presents an administration problem, since users need to be able to build big numbers of ships and position them intuitively around the map. A second issue is combat were scaling the resolution of the simulation becomes impossible (tracking the position of every ship, shooting every individual weapon, tracking the status of every shield, etc.).

Most games reduce this complexity by artificially limiting the number of ships that can be included in a ship, to some very small number (less than 20 in most cases), alluding to some "logistics limit". Others allow more ships, but user interfaces become tedious and unwieldy (Moo2). On both cases, the solutions seem kind of limiting and they take some of the feeling of grandeur that comes from being the sole ruler of a massive galactic empire.

The oldest and best solution is to group many ships into a single unit, adding up their capabilities to bigger totals. In this way, we don't model every independent unit, but model the group itself. For instance, a group of 100 ships with 2 lasers each just fires 200 lasers as a single attack, and the effects are calculated mostly as a probabilistic distribution instead of applied to individual targets.

This stacking together of identical units and keeping global stats is the centerpiece of this design, and it influences many of the choices made in terms of how ships are modeled, how the UI is behaves to display information and handle units and how combat is simulated. Any solution in this direction will hence need to be able to work equally well for stacks consisting of a single ship, to many hundreds of them, without impact either on usability or performance.

## Basic Combat - Minimum Viable Game

There are many ways to represent ships in combat, and they can get fairly complex to implement. Complexity can come from different ways to model units, different ways to calculate combat resolutions and the amount of interactivity and interfaces developed to support both. The next is a very minimal version, such that the game can be released and played. Ideally, further development would enable to customize and configure deeper ways to play the game.

### Ship Attributes

* **Hull:** a number that identifies the amount of damage that can be taken from an individual ship. In general bigger ships can tolerate more damage, but some small models can be heavily reinforced.
* **Attack:** A number that determined how hard a ship strikes an opponent, it is directly subtracted form the hull value of an opponent.
* **# of Hits:** most ships strike once, but some shot multiple projectiles or otherwise affect complete areas.
* **Initiative:** Initiative determines which ships strike first.

### Combat

Stacks are ordered by initiative. In order, each stack chooses a target from the opponent fleet and attacks it, causing damage. Once all stacks have attacked, a new turn begins in the same order. The combat begins when no ships are left. At the beginning of each turn, each player is given the choice to retreat from combat. If this option is taken, ships will leave instantaneously instead of attacking, whenever is their turn to take an action.

### Available Ships

* Star Fighter: basic small ship, inexpensive, single hit, decent initiative. Good against other fighters, lancers.
* Lancers: weak, best initiative, hard hitting, pounding large targets with powerful lasers from large distances. Good against Battle Cruisers.
* Battle Cruisers: huge, though ships. Average initiative. Have multiple medium cannons. Good against pit-bulls.
* Pit-bulls: medium and tough close-range fighters, using shrapnel to cover areas with lead. Low damage, but multiple attacks, very poor initiative. Good against fighters.
* Colony Ship: does not attack, can colonize new systems.
* Ground Troopers: Assault ships, for landing troops on enemy planets. Capable of invading a system.
* Bomber: terrible initiative, weak but pack a punch. Not particularly effective in ship combat, but extremely so when bombarding systems. Specialized bombers do twice as much damage as any other ship, and also provide a much better tactical bonus to assault troops.
* Defense Crew: a crew of tactical engineers and demolition mans that lay mines, deviates asteroids and creates automated defenses. Powerful when defending, but poor attackers. Good against all types of attackers.

## Advanced Ships - Long Term Development

### Ship Attributes

Each ship includes a fixed set of attributes, which varies from one model to another.

* **Hull:** a number that identifies the amount of damage that can be taken from an individual ship. In general bigger ships can tolerate more damage, but some small models can be heavily reinforced.
* **Absorption:** a number that directly reduces any damage taken by that amount, it summarizes any kind of defensive countermeasures, including shields.
* **Speed:** a number that determines how much a ship is able to move during combat (not in hyperspace).
* **Evasion:** ability of the ship to avoid being hit by enemy fire. It includes factors like ship speed, maneuverability, size, cloaking technologies, etc.

### Weapon Attributes

Each ship can contain one or more weapons, each one with the following attributes:

* **Optimal Range:** maximum distance at which the weapon can be used effectively.
* **Range Threshold:** total range difference till the weapon has no effect. A weapon's effectiveness decreases linearly from the optimal to the threshold, either because of power considerations or targeting considerations.
* **Accuracy:** a measure of the capabilities of each weapon to be aimed at a specific target. Better computers, faster actuators, mobile turrets and reflector targeting fields all increase the ability to hit other targets. Some weapons have a special accuracy of 0, which identifies a weapon that is not aimed against an individual target, but covers an area instead.
* **Damage:** total damage caused by the weapon. Can be fixed or variable.
* **# of Hits:** most weapons strike once, but some shot multiple projectiles or otherwise affect complete areas.
* **Effectiveness:** a series of flags (bit vector) indicating potential usefulness of the weapon in distinct scenarios. A weapon could for instance be usable in both space fights and for orbital bombardment, but another might not.
  + Space
  + Carpet Bombing
  + Tactical Bombardment
  + Invasion

### Factory Attributes

Each ship can contain one or more "factories". Factories produce special types of attacks that are considered more similar to ships than weapons. They usually travel at specific speeds, can be intercepted and might even switch targets or continue attacking after hitting the primary target. Examples include missiles, space troopers, assault shuttles, small fighter clouds, torpedoes, etc. Factories, apart from the attributes of the units they produce, have their own attributes:

* **Number of units:** how many units are created each time the factory is activated.
* **Number of charges:** how many times can new units be created.
* **Period:** how much time needs to pass between activations. This can be considered the time to recharge, reload, or prepare another launch.

Factories are particularly slow for direct attack operations, as they have to slowly deploy ships. Factories are best used for defense, as they can have units deployed on patrol for quick interception. In combat terms, factories can have a certain number of groups pre-deployed at the beginning of the combat, but only for a star at which they arrived on a previous turn.

### Special Attributes

There are some other effects that might need to be considered that differ from simply passing damage. Some weapons or units might write their own routines to calculate the damage taken and the damage caused, so that more complex behaviors are taken into consideration. Some examples include:

* Repair drones, which cause negative damage to your own ships.
* Assault shuttles, which steal some of the ships and create a new stack.
* Etc.

### Example Ships

The options described above provide a lot of flexibility in terms of gameplay. They can easily be used in both graphical and non-graphical battle simulators or even in interactive mini-games where battles are resolved using actual tactics. They also allow for the creation and design of custom ship models, a very common feature in many 4x space games. Nevertheless, a basic implementation might include a restricted set of existing ship types, for example:

#### Mechanic

* Lancers: weak and static, but pounding large targets with powerful lasers from large distances.
* Catapults: long range missile launchers, using deep-penetration nuclear warheads to crush large targets.
* Pit bulls: small and tough close-range fighters, using shrapnel to cover areas with lead.
* Carrier: (factory), deploying huge numbers of melee robots to swarm over enemy ships.
* Decoys: small ships that mimic other ships signatures, but can usually avoid damage.

#### Crystal

Crystal ships use lasers and are immune to lasers and lightning. Vulnerable to concussion damage. A ship, once destroyed, fragments into a set of smaller ships, and so on until they are destroyed at a bare minimum size.

#### Biological

* Minion: basic melee unit (small figther, not a buildable unit). Fast.
* Hive: A carrier for minions.
* Spore Hulk: A though-armored slow hulk. Its attack launches a set of explosive spores, similar to shrapnel. Minimal damage but over large areas. Good versus small ships.
* Arachnid:
* ? : though hulk, creates an immobile cloud that disrupts aiming, brings partial evasion to any ships hiding in the cloud.
* Scavengers: ships that collect biological remains from
* Seed Tree: great for defense, terrible attacker. Deploys enormous amounts of explosive seeds in the system, which lay dormant till a threat is detected.

#### Electromagnetical

* Etc.

## Space Combat

### Engaging the enemy

For a combat to happen, at least two hostile fleets need to find themselves orbiting the same star. By default each empire will try to control and block a star system, engaging any non-friendly fleet that comes to it. When this happens, a conflict will be generated, with all fleets in the system participating in it (from two or more empires). A simulation will be performed, either interactively or automatic, to calculate the outcome of such a conflict.

During the simulation, each fleet will choose potential targets from all the fleets with a trust level inferior to "cease fire"[[2]](#footnote-2). If there are no such targets, the simulation ends and the conflict is solved.

### The Combat Turn

The combat plays as a simulation on a single dimension. Both fleets start at a random distance from each other, which can be anything from close range to 2 times the longest weapon's range. It is then more likely that fleets will close on each other gradually, but in some occasions things can get how quickly!

Each turn, smaller units will be launched if possible, ships will move forward at their characteristic speeds, and weapons fired. These steps occur in this order, for all participating fleets simultaneously. After they have all been resolved, a new turn will start.

The battle will only last for 20 combat turns, after which the game will continue normally on the main map. This allows players to retreat, repair or reinforce, but if at the end of the game turn fleets are still in the same system, they might engage again. In a sense, small battles will probably be resolved in a single game turn, but larger conflicts will last for several turns, with opportunities for negotiation and even sabotage in between.

### Choosing Targets

The damage a weapon is able to cause to the enemy is calculated independently for each possible pair of attacker and defender, using the following formulae:

In essence, the weapon's efficiency will depend on how optimal is the range, how easily it can hit the target and how much it can overload the target's defenses. Note that in the case of area weapons it is considered that targets cannot evade successfully, so accuracy is not considered. Nevertheless, part of the damage is bound to be dissipated in empty space, so area weapons are not completely effective.

On any given turn, potential damage is calculated for every weapon and target, and target chosen according to where is the maximum damage inflicted. This very simple heuristic does not consider more complex eventualities, like how killing a specific enemy first might minimize the damage to another, friendly stack of ships. These considerations might easily explode in both complexity and uncertainty, so the simplistic option is preferred. The simple approach is also more understandable for players.

### Applying damage

One important issue when ships are aggregated together is how to measure the amount of damage the stack has received. One common approach is to model the remaining life of each independent ship, but this would not scale and thus break our design criterion.

Another alternative is to apply all damage to a single ship in the stack (done in Moo1) and when that one is destroyed start applying damage to the next. This simplifies things considerably, as it only requires storing the current level of damage for the one ship, plus the total number of remaining ships. There are a few problems with this approach. First, it is unrealistic that weapons will be able to be aimed perfectly to a single target at the time in the middle of vast clouds of ships. Not only would some ships be badly positioned to concentrate fire to such extent, but it is also to be expected that damaged ships would retreat or find cover behind their allies. The second issue has to do with gameplay; applying the damage in such an efficient way will immediately diminish the stack strength considerably, making big ships more attractive overall (since they can sustain more damage and stay operational) and thus invalidating some interesting strategic options. It also makes bigger ships more likely to survive a battle, albeit in need of some repairs. Lastly, it is not very compatible with area weapons, which by definition should attack several ships simultaneously.

Our solution is to apply different types of damage in different ways, but to do so in a framework simple enough that calculations are easy and scale without problems and realism does not suffer too much compared to the full-fledged simulation. We keep only 3 variables:

* Number of ships
* Base damage
* Maximum damage

The idea is to represent damage to a stack as an area. The number of surviving ships being one dimension and the damage taken by them another. The total hull of the stack is then the multiplication of both units. As an example, let's say that we have a stack with 100 ships, with 6 HP each and that it takes 400 points of damage from a single attack. Each one of the three gray regions below, all with equivalent area, could be used to represent that damage:



In the first representation, damage is evenly distributed among all ships in the stack. No ship is actually destroyed, but they all become severely damaged and vulnerable to new attacks. This is a good representation for area attacks.

In the second representation the damage is distributed among only 50 ships, losing 8HP each. Now, since they only had 6HP to begin with, they are all destroyed, and the excess damage is lost. This is what we described before, but it matches well the case were a focused weapon of great caliber hits a small ship. We expect it to be immediately destroyed and we don't expect other ships around to be affected. So, if the damage per hit of a focused attack is higher than the HP per ship, we just diminish the size of the stack by the number of hits of the attack.

The third scenario is when focused weapons are brought to bear on a more resistant stack, were multiple hits are needed to bring down a single ship. In this case we would expect the attacker to try to focus its fire on a few targets, but we would also expect the defender to shuffle and protect his wounded. As a result, a good compromise is to decide that some ships will receive a lot of damage and others little, depending on how many are there to shuffle around. The triangle represents a linear distribution of this effect, we the less damaged ship takes no damage and it grows from there. We know that the most damaged ship would take twice the average damage, which in this case corresponds to loosing 8HP, and the maximum is updated by this much, leaving the base damage intact. In this case, we can see that 25 ships took too much damage and were destroyed, with some extra damage lost. The risk of overkill is always present when we concentrate fire.

The nice thing about areas is that then can be easily added up when successive attacks need to be considered. Consider this graph, were each color represent the damage done by a separate attack:



Note how area damage just upgrades the base damage, and focused damage can easily be applied on top, by adjusting the maximum damage. Every time this happens, the diagonal line is intersected with the HP limit, and the number of surviving ships updated. The graph on the right shows what happens if high focus damage is applied when some damage already exists. We can't consider all that damage to go to already damaged ships nor to undamaged ships. Neither is fair. Hence, we just modify the number of ships without altering the maximum damage, squeezing the distribution. This is equivalent to say that we select ships to be destroyed fairly from all positions in the damage curve.

As it can be seen, all these operations are not only nice and fair representations, but also extremely simple to compute!

### Choosing strategies

In this simplistic combat model, the only real choice a stack has to make is how to move. How much should it advance or retreat? While advanced AI can be created, some simple options include:

* Always move forward, till the middle is reached. Easy to implement and understand.
  + Probably decent results, with initial stages of long range, followed by close range fights.
  + A good initial implementation for testing.
* Only move forward till there is at least one target at optimum range.
  + May fire sub-optimally.
* Move the entire fleet to the maximum range of any weapon.
  + May waste important firepower.
* Always try to move in the expectation that damage done is maximized. That is, calculate not only which stack to hit now, but also which stack would take more damage if range were ignored. Move so that that target is at optimal weapons range.
  + Of course, recalculate every turn, as the most convenient target will change.
  + This criterion is very decent. Not as aggressive as always advancing, but well-tuned for an aggressive, all-out fight were each stack behaves close to optimal.
* As before, but consider the damage a stack would take if it closes (what enemy stacks would chose it as the enemy) versus how much it would inflict. Advance or retreat depending on this difference.
  + May take vulnerable units out of the fight, saving them from destruction.
  + Retreating stacks might leave other friendly stacks to die.
* Imagine that all stacks in each fleet move together. Sample a set of distances between them and calculate relative damage. Move units so they try to get to that distance on average.
  + Could produce funny artifacts.
* Advance till a first round of damage is computed. If the fleet did more damage than it received, all units keep on the offensive, else retreat.

These are only examples of possible strategies. The first and fourth strategies are probably the best ones, since any strategy that chooses to retreat a stack might be causing unpredictable consequences to other stacks. Also, any strategy that is not greedy in terms of the damage caused might be easily countered by the enemy by creating fleets that have tough, evasive units in front and light but hard-hitting units a little back. Of course, many of these strategies might be tried, or even chosen depending on fleet composition, race, etc.

### Retreating

The adversary can choose to retreat from battle. In this case their ships move away from battle and escape after getting out of range of any enemy fire or after 3 rounds, whatever happens first. At least 1 round of combat is always performed.

Retreating ships do not attack, but they can still take damage.

## Other

#### Blockades

If at enemy ships orbit a star system unopposed (other fleets are not present or are avoiding confrontation), the systems is blockaded. Such a system may continue to grow unimpeded, but 1% of any production that goes into the reserve will be lost for each blockading ship. Bigger ships or those with more weapons might count more in the future, but the idea is a certain number of ships to hunt commercial freighters. The Death Star, no matter its size, is probably not the best ship to create a blockade.

#### Control

Fleets are useful to control territory. Even if they do not engage in combat and allow other fleets to orbit the same system. Fleets in a system prevent any other fleet in the same system to jump in the direction of TODO???

## Conquest

### Troops

### Invasion

#### Collateral Damage

# Intelligence and Influence

### General Description

In most existing 4x turn-based strategy games influence, diplomacy and trade is handled completely outside of the main map. It is a separate endeavor, with little resemblance with the game mechanics present for conquest and little (if any) geographical interaction. In contrast, our approach involves using game units to handle these aspects of the game, using them in a separate but consistent way that closely resembles the way military units are handled.

The goal is to have influence units moving along the main map, traveling in the same way that military fleet move and being placed at specific and strategic locations. This not only enriches and deepens the influence game, but also allow for greater interaction and conflict between the use of force and the use of cunning, being both potentially equally viable strategies that can also be mixed together.

### In enemy territory

#### Infiltration

Influence units can be moved between stars in much the same way as normal military ships, but the interaction between them is limited. Influence units are in general considered hidden or undercover, but every turn there is a chance that the might be discovered:

This means than in a fully closed relationship a unit should expect to be stay undercover for 20 turns on average, where in a fully open relationship they might stay hidden basically forever. If an agent is found, he still has a 50% chance of automatically retreating to a friendly star system before being destroyed.

Troops stationed on a system might also discover infiltrators, having a second (and completely independent) chance of finding enemy agents given by:

Note that this probability can be greater than one, in which case multiple agents are found. For instance, if the probability is 2.7, 2 enemy agents are found and there is a 70% chance of finding a third one. For this to happen though, a system needs to be heavily militarized. Agents found by troops also have only a 10% chance of retreating.

#### Sleeper units

An agent in enemy territory can go deep undercover in order to avoid being found. Such an agent can't be found, but can't perform their original missions either or move from one star to another. These sleeping cells can be activated at any time, resuming their normal functions.

### Units

#### Infiltrators

Infiltrators are the basic agents that go into enemy territory. They gather intelligence about the star they are deployed in and form the backbone of the intelligence network inside the foreign territory. This network is required to support undercover actions, sending orders and resources to the agents in the field.

In terms of intelligence gathering infiltrators allow visibility of what happens in the outpost they infiltrate, much in the same way than having a fleet in orbit but without risking battles or damage relationships.

In terms of the intelligence network infiltrators need to be connected by a single starlane to another active infiltrator, all the way to one or more of their own outposts. Other agents can only travel into foreign territory if an active cell has already been established in the desired location. This limits the expansion of agents deep into foreign territory and increases the risk of detection, making the infiltration game much more challenging and interesting.

It can happen that a cell is detected or goes undercover, breaking the link and stranding other agents in enemy territory. Those agents cannot go undercover and have to start retreating automatically to the closer infiltrator still connected to the network. Luckily, infiltrators are always the last agents to be eliminated from a star system, so redundancy is extremely important.

#### Smuggler /Trader

A smuggler is a unit that specializes in intervening markets for profit. Each smuggler in an outside empire will generate 0.01 bc of production per turn from commerce with that empire, up to 20% of the current production of the system in which they are stationed.

If the trust level between the empires is such that trade is allowed and legal, smugglers get several advantages:

* Their efficiency and maximum caps double.
* They generate half as much additional production to the other empire as they do for their own empire, creating a symbiotic relationship that also improves relations.
* In case of discovery they remain visible to the other empire, but are otherwise not disturbed at all while the trust levels stay high enough.
  + A trader that has been discovered only becomes a smuggler again after re-entering their own territory, were they can be given new identities.
  + A trader is immediately discovered and killed if he finds himself in an enemy empire when the trust level goes below the trade threshold.

Smugglers can never get access to the well-guarded special resources owned by other empires, but with enough trust they can, just not while undercover. Up to 3 smugglers can willingly become open traders in order to each get 10% of the bonus given by the special resource. This is an important event, so both empires are notified.

#### Spy/Scholar

A spy can try to get the secret technologies developed by other empires. A single spy produces a base of 1 research points for each technological area, plus 2 additional research point in each area in which the foreign empire has a higher technological level. A single outpost can't produce more than a number of research points equivalent to 20% of the outputs production.

If the trust level between the empires is such that cultural cooperation is allowed and legal, spies become scholars and get several advantages:

* Their efficiency and maximum caps double.
* They generate half as much additional research points to the other empire as they do for their own, but adjusted according to the inverted difference between technological levels, such that a scholar always acts as a homogenization force.
* In case of discovery they remain visible to the other empire, but are otherwise not disturbed at all while the trust levels stay high enough.
  + A scholar that has been discovered only becomes a spy again after re-entering their own territory, were they can be given new identities.
  + A scholar is immediately discovered and killed if he finds himself in an enemy empire when the trust level goes below the trade threshold.

#### Privateer

The privateer (or pirate) is a unit designed to harm an enemy's economy. A privateer raids the systems were he is stationed, decreasing the system's production by 10 bc each turn. There is no real limit to the number of privateers that can be effective simultaneously, so an outpost production can be eventually brought down to zero. On the other hand, the bigger the number of privateers operating in a system the harder it becomes to hide their operations, so the probability of discovery is multiplied by half the number of privateers.

Privateers are discovered by fleets instead of troops, using the same formulae. If discovered though, privateers are not automatically eliminated, they put up a fight. They are not formidable units but can hold their own against small units and cause some damage when cornered, maybe even surviving the battle.

While some infiltration is accepted and other frowned upon, a privateer's path of destruction causes extreme damage to trust between empires and even war. Privateers are rigged to autodestruct if possible, leaving no recognizable trace of their origin, but after destroying a privateer there is still a 50% chance that the foreign empire will learn its origin. If not, normal pirates will be blamed.

#### Assassin

An assassin is an agent trained with the specific objective of finding and eliminating other agents. When in local territory, the assassin has a 10% chance of finding an enemy agent and killing it. When operating in foreign territory, it has only a 5% chance of doing so. Assassins are essential to the intelligence conflict, both in a defensive and in an offensive position.

#### Saboteur

The saboteur is a little different than other agents. It does not produce anything each turn, but it slowly prepares for the eventual unleashing of massive destruction. For each saboteur and each turn a sabotage point is produces and carefully stored, but no saboteur can prepare more than 10 points before executing sabotage. If a saboteur is moved, the preparations are lost.

Saboteurs are most often activated directly, the player or AI choosing the right moment to strike against either:

* An outpost infrastructure.
* Fleets or troops stationed in the system.
* Research efforts (limited by the outpost production and the technological expenses of the foreign empire).

#### Corruptor

Corruptors slowly but surely create dissent in the chosen system, generating propaganda turn by turn such that eventually the whole outpost revolts against its former empire. The speed of this process depends of course on the total infrastructure of the target outpost, and can never be less than 5 turns.

Corruptors are the most exposed agents, and as such are always chosen first if when agent detection happens.

#### Diplomat

Diplomats give resources away in order to improve relationships with other empires. The resources are given as a token, improving the growth of systems were they are stationed. If the systems can no longer grow then diplomats stop having an effect. While diplomats can affect no direct control over empire relationships by human players, they can change the rate in which society follows the trends established by the player, making it as much as three times as slow (depending on the impact of donation compared to the other empire's economy). In any case, both human players and AI may look favorably on such donations.

# Technology (TODO)

Similar to the way it’s handled in Titan Quest, assigning levels in a main branch and allowing special projects at certain levels. Special projects should in some way change game mechanics instead of being just bonuses, for example by allowing new forms of travel, new intelligence options, new things to be built, etc. Branches can include:

### Space Warfare

* + Ship Attack
  + Ship Defense
  + Troop Attack
  + Unlocks
    - Specials
      * Shipyards
      * Barracks
    - Ships
    - Invasion troops
      * Marine Pod -- basic invasion unit
      * Commandos -- decreased collateral damage
      * Void Commandos -- able to fight in space combat
      * Tactical Bombers
      * Deep-Penetration Missiles
      * Orbital Punchers
      * Mass Destruction Weapons

### Xeno-Domination

* + Trade/Smuggling Efficiency
  + Infiltration Efficiency (hiding/moving through enemy territory)
  + Cover Ops Efficiency (espionage, sabotage, intelligence)
  + Unlocks
    - Specials
    - Units
      * Spy
      * Saboteur
      * Trader
      * Smuggler (worse than a trader, but works during war, low trust)
      * Privateer (uses warfare to steal)
      * Assassin
      * Politician

### Development

* + Construction Efficiency (cost of new infrastructure and units)
  + Research Efficiency (cost of new research)
  + Maintenance Efficiency (cost of unit and infrastructure maintenance)
  + Adaptation Efficiency (tolerance to bad star conditions)

### Exploration

* + Sensors (perceive ships on hyperspace)
  + Fleet speed / Jump Distance
  + Unlocks
    - Hyperspace Gates
    - Hyperspace Tunnels
    - Hyperspace Scanners
    - Hyperspace Dampeners
    - Hyperspace Communicators
    - Jump Drive
    - Jump Gates (
    - Omniscient scanners (discovers planets and fleets)

# Story (TODO)

## Race

### Concepts

* It is important that language/commerce/infiltration/diplomacy seem reasonable between races.
* It is important to somehow standardize on the kind of resources each race wants, as to avoid a complex structure with "per race" star definitions.

### Preliminary Ideas

* Purebreeds (standard humans): Focus: balanced
* Androids (those who perfected mechanical forms, beautiful, perfectionists). Focus: influence.
* Cyborgs: industrialists. A race created and used on early-colonization. Centered on getting work done, survival. Focus: industry.
* Crystaloids (crystal-empathy, use crystal ships with liquid crystal compartments). Very adaptable. Extensive use of lasers. Support high pressures. Slow growth, but high adaptation. Movement bonus inside empire due to resonance. Focus: expansion.
* Plantoids: ships are trees. Have a plant affinity. Very efficient photosysntesis and very low growth requirements and maintenance costs in general. Generally pacifists, life lovers, nurturing planets. Focus: growth.
* Trascendents: Psilon-like brains extended by alien drugs. Focus: Technology.
* Singers: highly telapatical race of nomads. Almost ommniscient understanding of the cosmos. Traders. Many living inside hyperspace itself. Very small outposts and production. Religious searching for the "song of the cosmos". Focus: mobility, trade, hit and run.
* Reapers: zerg-like humans, beyond all empathy. Focus: extermiantion, overconsumption.
* Moles: product of a tragic history of survival on a high-G rock planet. These humans are small and inhabit deep underground complexes. They resent ships so usually retrofit big asteroids for travel. Slow but tough. Focus: invasion.

## Storyline

* Mankind explored space for centuries and as it did it adapted and mutated in different ways.
* Other sentient life was found, and a wave of xeno atraction, plus the discovery of empathic links allowed humans and other species to interbreed. Several sub-races were thus born.
* But deep down humanity, while diverse, remained the same. The was war, there was peace, there way joy and sadness.
* Till one new sentient being was empathized with… and the reapers were born. Only war was available then.
* The reapers consumed and expanded like a virus, leaving nothing behind but desolation. There was no stopping them.
* The variants were thus unified by need, and they pooled their resources to be able to flee. On a huge experiment that caused a supernova, space and time was bent so a few arcs could flee to another galaxy, a new galaxy.
* … but the reapers followed.

# Roadmap

|  |  |  |  |
| --- | --- | --- | --- |
| Modules | High Priority | Normal Priority | Low Priority |
| Universe | * Basic structure ✔ * Static configuration ✔ | * Point generation ✔ * Triangulation ✔ * Edge generation ✔ * Min. span tree ✔ * Area-based pruning ✔ * Saving/Loading | * Plugin architecture * Procedural backgrounds * Star fields and parallax |
| Stars | * Random Star Stats ✔ * Fleet orbits ✔ | * Animations * Star database load | * Header images and story |
| Main controls | * Zoom in and out ✔ * Move with cursor on borders ✔ * Zoom levels ✔ * Coordinate Mapping ✔ | * Main menu |  |
| Economy | * Production ✔ * Expenses collection ✔ * Economy Mayor Controls ✔ | * Strategic resources |  |
| Empire | * Name and color ✔ * Pseudo-random initial placing (1 hop prohibited) ✔ | * Empire relationships * Empire UI * Comparison Graph | * K-medoids clustering for placement ✔ |
| Fleets | * Fleet UI ✔ * Fleet basic movement queue ✔ * Basic design (ship id + number), no particulars ✔ | * Long-range movement (IDA\* search) * Define several predefined ship types | * Custom ship components * Custom ship design window |
| Shipyards | * Construction ✔ * Shipyard UI ✔ * Unit production ✔ | * Upgrades | * Relocation? |
| Influencers |  | * Spies |  |
| Tech |  |  |  |
| AI | * Some basic AI to play against. |  |  |
| Audio | * Basic Music * Basic click sounds | * Music | * Music and Sounds |
| Game Mechanics | * Turn mechanics ✔ * Conflict resolution ✔ | * Fog of War ✔ |  |
| Combat | * Simple solver (mimic Vox) ✔ | * Linear Solver (automatic) * Animation * Adjustable simulation speed | * Linear with real-time commands * Non-Linear? |
| Invasion | * Automatic on fleet arrival ✔ |  |  |

# Turn & Event Processing

## State

### Concepts

#### The true state

This can be considered the shared, true state. It includes everything going on in the galaxy.

#### Visibility

Visibility corresponds to one or more locations on the true state, visible to a given empire. Locations correspond to stars and lanes. Visibility is limited to those the current location of a colony or unit (either lane or star) and if orbiting a star also those lanes who reach it. This is a good and practical tradeoff, which also plays well with the fact that we want hidden agents to play an important role. There is no such thing as sensors capable to detect things several light-years away, which is also more realistic.

#### Perceived State

This is the state of things as perceived by a single player/empire. The perceived state is a mix of the things that were visible (some of which can be remembered), the things that are currently visible and new actions taken by the player this turn.

#### Decisions

A player, by analyzing the perceived state, need to decide what to do in order to win the game. Possible decisions include queuing new units to be built, colonizing a place, starting projects on a given colony, moving task forces around, etc. Decisions will in turn issue commands to the relevant entities or create new entities. These changes exist immediately, but no progress will be made till the turn advances.

#### Events / Options

As a result of previous decisions (or randomly), diverse events can take place. Some events involve immediate consequences, like two fleets engaging in combat. A second set of events will have no direct consequence, but will enable or disable a player from making certain decisions, for instance, a colony ship leaving a star will remove the option to colonize it.

Events have a given precedence. For instance, the option to colonize cannot be determined before space battles are resolved, as the colony ship might be destroyed.

### Procedure

* Pre-turn phase
  + The true state is updated (time passes, production occurs, fleets move, etc.)
    - A list of changed locations is gathered. This is done to avoid checking all possible locations, especially in large galaxies. Note that events cannot be created immediately, since they migh depend on updates not yet made.
  + Stored places are checked for events in the right order. Global events are also computed (disasters, etc.).
  + Events are resolved
  + Visibility is recalculated for all empires
  + Perceived state is recalculated for all empires, by updating the remembered state with the visible locations.
  + All transient things (fleets) on the remembered state are removed from non-visible locations.
* Turn phase
  + The player or AI observes the perceived state and takes actions.
  + Actions change the perceived state only (fleets created or disbanded, routes modified, etc).
* Post-turn phase
  + The changes made by each player in their own perceived state are applied to the true state, one location at the time.

### List of immediate events

These are events that produce immediate resolution. Once the event is triggered, outcome is processed (possibly stopping turn processing in the meantime) and no permanent object is created.

* Fleet merge
* Space combat
* Bombardment
* Invasion

### List of Options

These are long-lived. Each turn they should check if the option is still available or remove themselves. While available, a player can decide to execute the option, causing an effect and potentially generating other long-lived game entities or options. Note that a human player might see the event take place immediately for display purposes, but other players will not see those changes until next turn.

Note that options are tied to specific locations, available to specific empires, and that it makes no sense to have multiple options of the same type there.

* Create shipyard / academy
* Colonize
* Sabotage

### Step details

An expanded list of the things that need to happen, and the correct order for them to happen, follows:

* Update
  + Clean up from last turn
    - Reset counts
    - Reset statistics
  + Update all fleets (of any type)
    - Movements
    - Expenses
    - Post Statistics
  + Update technology
    - Status
    - Expenses
    - Post Statistics
  + Update shipyards
    - Production
    - Expenses
    - Post Statistics
  + Update all outposts
    - Production
    - Expenses
    - Growth
    - Post Statistics
* Conflicts (discovered by location)
  + Merge fleets if needed
  + Check for sabotage
  + Check for space combat
  + Check for bombing
  + Check for invasions
  + Clean empty fleets
* Orders
  + Launch AIs
  + Open the turn for player input and instructions

As many of the items as possible are solved automatically, asking for user input only if necessary. At any point in time, only a single event is active, and they are processed in order. If an active call detects that the current item is solved, it moves to the next item currently in the list or to the next location. If the item is marked as requiring user input, then the loop is stopped until the event is again marked as complete.

# Interfaces

In general: allow using space to hide the control, in order to be able to create routes or check the map easily.

## The star control

### Overview

### Actions

While most actions depend and/or require specific units to be in orbit, the options are part of the star interface.

* Base
  + Colonize (available on unclaimed stars with a outpost ship present. Generates an event only on arrival)
  + Build (available on owned outposts, but not an event)
    - IHQ
    - Hyperspace Gates
    - Hyperspace Tunnels
    - Hyperspace Scanners
    - Hyperspace Dampeners
    - Jump Gates
  + Military actions (available with the correct unit in orbit, generates events every turn)
    - Bomb (once per turn)
    - Invade (once per turn)
  + Influence actions (available with the correct unit in orbit)
    - Sabotage (activate manually, once per turn)
    - Trade (toggle manually)
    - Espionage (toggle manually)
    - Corruption (toggle manually)

Things to take into consideration:

* Actions that last more than one turn, requiring saving state.
* When do actions become available (check on click v/s check on turn)
* Different actions for different empires.
  + Especially the player empire.

## The fleet control

* Buttons
  + Toggle All / None
  + Invert selection
  + Disband selection
  + Leave in orbit
  + Toggle auto-merge for selected fleet
    - Confirmation dialog.
    - If set to auto-merge, this happens instantly.

## The IHQ control

* Buttons
  + Toggle full output / half / none
  + Clear queue
* Relocation (click on star)

# AI

Note: Inspired in no small part by the design presented here: <http://www.gamasutra.com/view/feature/129959/designing_ai_algorithms_for_.php>, plus some AI elements present at most turn-based games.

## Pre-requirements

* Path and distance calculations: There needs to be an easy way to map a task against a fleet or object to execute it. Distance is important to determine the best object to execute the task.
* Visibility: tasks cannot be created in cases where empires are not aware of them. The AI needs to be able to quickly evaluate the visibility of stars and fleets. Visibility may involve remembering something seen in the past.

## Production Assignment

The first part of the AI is the allocation of production for distinct purposes. Resources can be assigned to each area (combat, technology, growth, agents, savings, expansion, exploration, etc.).

The algorithm will be based on incentives. The events and circumstances of each turn will produce more incentives to one area or another. Each incentive created has a weight, which will then decay over time at a specific rate for that type of incentive. Some example incentives are:

* An intrinsic tendency by an empire to prioritize growth over expansion, which is constant and does not decay.
* The fact that there was not enough shipyards to spend the production assigned to units, which will decay over 5 turns.
* The fact that another empire is weak, and could be obliterated for his territory.
* The fact that another empire is much more advanced technologically, so the AI needs to catch up.
* The fact that there is a nice star that can be colonized.
* The fact that very little of the galaxy is known, so we need to explore more.
* Etc.

In fact, incentives can be very varied, and come from different sources. Also, recurring events that continue over many turns would produce several incentives of the same type, so the AI does not seem to react only to short-term stimuli. This has to be carefully tuned along with decay rates to also avoid some concerns to become overwhelming.

Once all incentives have been created, they are added up. This in turn produces a series of positive or negative values assigned to each area where resources can be allocated. Negative numbers are converted to zero (we don’t actively destroy previous investments) and the remaining positive values are normalized to obtain percentages. The corresponding amount of production is then allocated to each area.

### Allocation amount

The amount of production to allocate on any given turn will depend on the urgency of the situation. In general, the amount of production that **can** or **should** be spent is proportional to what is being invested, and it makes sense to grow this capacity organically turn over turn. We don’t want to run our reserve to the ground nor to put all our production in savings.

The AI could by default spend the entire turn’s production, multiplying this by a value that would depend on how many turns of production are currently in the reserve and what was the total value of all incentives before normalization (a measure of the urgency of the situation). This will keep the reserve at a reasonable, but low amount.

### Overshooting and undershooting

In many circumstances, the resulting allocation by any given category might turn out to be too little or too much. In these cases, the AI can react by changing the internal strategy pertaining to a given area to adjust to the difference, while generating additional incentives that can correct resource assignment in the future. Examples are:

* Too much money is assigned for growth, but it could not be spent under the current configuration. The AI might as a result decide to enable the use of the reserve to boost production. If it is already enabled, it might decide to increase the return of investment limit. Separately, it might produce a disincentive to future expenditure on growth, which would affect the allocation next turn.
* A significant amount of money is assigned for ship building, but there are not enough shipyards to spend it all. Building a new shipyard is an expensive proposition, and should be postponed if possible. The AI might create a small incentive to receive more funds for a new shipyard. Over many turns, unused allocation to ship building can increase, up to a point where it can pay the per-turn expenses of the project. When this happens, a shipyard is built or upgraded (which one is a separate analysis by the AI).

There might be cases where several areas are underfunded, but since they would all produce additional incentives the assumption is that the normalization step will prevent extreme competition. This behavior is highly adaptive and might fluctuate from turn to turn, but it will hopefully remain close to an optimum.

## Production Choices

What to produce? Very dependent on particular logic for each area.

## Strategic objectives

* High level goals
  + Survival
  + Victory conditions
  + Overall superiority
  + Intrinsic tendencies
* Knobs
  + Empire relationships
  + Hostility
  + Helping others
* Determine on this phase:
  + Update empire relationships.
  + Who is the main adversary?
    - Do not take on many adversaries at once unless inevitable.
    - Generate potential events.
* Generate categorical comparison with other empires
  + Calculate comparison matrix
  + For every module, how well am I compared to others?
    - Tech levels
    - Influence
    - Number of planets
    - Production
    - Fleet size
    - Unit production
  + Trigger unusual actions for severe differences
    - Ask for help
    - Surrender

## Object Classifications

For the purpose of a single empire, stars and fleets are classified as follows:

* Ownership & Visibility
  + Unknown: stars not visited, but assumed to exist at the end of existing lanes.
  + Unclaimed: stars not owned by any empire.
  + Owned: stars or fleets owned by the empire.
  + Friendly: stars or fleets owned by empires with Brotherhood status.
  + Unfriendly: stars or fleets owned by any other empire.

## Tactical Objectives

* Unit Actions
  + Generate list of tasks for existing units
    - Military Defense
    - Influence Defense
    - Military Conquer
    - Influence Conquer
    - Expand
    - Trade
    - Repair
    - Explore
  + Compute optimal units to assign to each task
    - Use distance as an important metric
    - Use relationships to weight them (defend, attack).
    - Use race/AI tendencies to weight them.
  + Aggregate the total satisfaction score for each action category.
* Global actions
  + Generate list of global tasks
    - Build more units
      * Evaluate independently for each unit (especially important for influencers)
    - Improve technology
      * Evaluate independently for each tech type
    - Colony growth
      * Score based on ROIs
      * Score based on maintenance versus production
    - Improve relations
    - Planetary improvements
      * More/Bigger HQ needed (determine by queue size, distance to frontlines)
    - Save money
      * This should be a fixed, kind of low score, so it would only take meaningful priority if all other tasks become low priority themselves (peace, abundance, no un-colonized planets, etc.).
  + Use race/AI tendencies to weight them.
  + Merge with satisfaction aggregates to balance against building new units.
  + Combine with cross-empire matrix to weight universal needs
  + Distribute wealth accordingly to each category
    - Apply money directly to global efforts.
    - Use original unsatisfied needs to determine what to build and where.
      * Don't create queues more than 10 turns long.
      * If extra credits are left, save for HQ investments. This should work for when there are no HQ left and also, provide a nice gradient between a slight need for an HQ over many turns versus an urgent need.

## Accumulators (code)

* Empire
  + All outposts
    - Economy
    - Defense
  + Empty stars
    - Colonization
* Calculate visibility 🡪 signal
  + Create perceived state
    - No need for classification (here)
  + Create AI tasks
    - Tasks on remembered state?
      * Need to store other state.
    - Classification required.

# Sandbox

## System Developments

While the main idea is to avoid the micromanagement associated with building things on each and every system, there are some special technologies that can be used to alter the "terrain", the way things interact on the main galaxy map. These are built from the map itself, by creating a global project financed at the empire level, but placed at a specific location. After selecting the system, options can be found to build these projects, indicating their cost and turns to be completed.

Ideally, system developments, as part of the galaxy landscape, should be seen on the galaxy map, and not hidden on a separate screen or menu. This need limits considerable the number of developments that should exist, as there is limited ways to visualize them without adding too much clutter. Also, only a single instance of each development can be built. A list of existing system developments follows, but details for each one of them can be found on the relevant parts of this document.

* Colonization: transforms an outpost into a full-fledged outpost.
* Shipyard Construction: creates a new basic shipyard in the selected system.
* Gates
  + Hyperspace Gate: creates a gate allowing task forces with no gate technology to jump into hyperspace.
    - Can be upgraded with accelerators
  + Jump gates: creates a gate allowing task forces to jump to a system with another gate.

## Turns

TTTV is a turn based game. Each turn consist on multiple phases, each restricted to operate over a specific set of events or behaviors.

* Event resolution: All conflicts and scheduled events placed on the last turn are resolved.
* State update: All objects are updated and the time moves forward.
* Order placement: All players (human or AI) plan and place orders that will begin to take place on the next turn (some can last for more than one turn).

### Possible Order

It is not trivial to define the order in which these phases occur. They can be in strict succession or some can even be calculated in parallel. Here are some thoughts on the matter:

|  |  |  |
| --- | --- | --- |
| Update 🡪 Conflicts 🡪 Orders | Update 🡪 Conflicts, Orders | Update 🡪 Orders 🡪 Conflicts |
| Separate stages for user input, forces narrow focus. | Temporarily Blocked Elements. Access to other state in between. | Smooth order phase. |
| Can choose on conflict based on outcome of previous conflict. | Can choose on conflict based on outcome of previous conflict. | Worse control of conflicts depending on other conflicts. |
|  | Can choose resolution order. |  |
| Easy to code. | Complex to code. | Trivial to code |
| Everyone wait till conflicts are resolved. | Resolution needs to wait till everyone decides to check. | Great for multi-player, as all orders are serialized and then resolved based on previous asynchronous choices. |
| Fleets go into combat on arrival | Fleets go into combat on arrival | Fleets go into combat at end of turn, possibly after receiving orders. |

It is not clear which one of these is best or if it would be a good idea to force users to solve all conflicts as soon as possible (it alters what him and other players, even AIs, see as the turn's reality). For simplicity and fairness and to avoid exploits, the first option seems like a good initial alternative, although some exploration is possible.

## Actions & Events

### Context

Actions are optional commands available to a player. These actions usually depend on several factors: location, presence of units or outposts, treaties with other empires, enemy presence, actions taken last turn, etc. Some things occur without direct command by the player, but as automatic events instead. Actions and events are fairly similar, so it makes sense to try to address them in a single structure. Actions and events may depend on specific modules, making classification and implementation particularly difficult. Some important observations include:

* Access patterns probably involve location and empire.
* Some actions are available by default for a specific empire, on all locations.
* Some events do not refer to a location.
* For those that do refer to a specific location, the UI might want to move the camera to center on that location.
* Some events need to take control of rendering, input and/or camera.
* Some events might block the game state till they are resolved.
* Most events are a result of fleets arriving at a system or leaving it. Since the number of arrivals should be much smaller than the number of stars, it makes sense to figure out events only in these locations.
* Also, some actions are only needed under user interaction, which happens even less and outside the event loop. An example is colonization, which does not happen automatically.
  + Note that AI might use global events to detect even the chance of doing these things, so there is something to say for automatic detection and caching.
* Events may last for more than one turn, and cease to be available due to other events (a fleet departing or being destroyed, etc.). Persistent events should be checked every turn.

### Proposals

Each time a turn ends, follow these steps:

* Run reset/initialization turn steps
* Run update steps on base objects (outposts, fleets, empires)
  + Collect locations to check for fleet arrivals
* Run event checks for new arrival locations
  + Remove any persistent event from these locations.
  + Generate new set of events for these locations.
  + Apply any effects in priority order.
* Check for updates on old event locations
  + Remove event no longer valid.
  + Apply any effects.

### Event list

As the game goes, there are many events going on, with very particular characteristics:

|  |  |  |  |
| --- | --- | --- | --- |
| Event | When | Automatic? | Location |
| Colonization | A suitable unit arrives at an empty star. | No | Unclaimed |
| Fleet Merger | A fleet arrives at a system which already has a ship of the same empire. | Yes | Any |
| Engage enemy forces | A direct command on a system with at least one fleet and an enemy fleet. | No | Any |
| System Project | Available on all owned outposts. | No | Owned |

Events

* Happen at
  + Star
  + Empire

Galaxy

* Triggered by:
  + User
  + AI
  + Turn
* Then progressing
  + Instantly (taking modal control)
  + On turn
  + Over several turns

Note that for AI purposes, all possible actions (or most of them) will likely be generated anyway at the beginning of each turn. AIs will not be opening interfaces. Also consider paralelism.

Check 🡪 generates action 🡪 event

## Spending

|  |  |
| --- | --- |
| Expense | Description |
| Outposts | Includes table with all outposts. Includes star stats, expected maximum production, cost of infrastructure, button to colonize. |
| Outposts | Includes table with all outposts. Includes production, last turn growth, current industry, expected maximum industry, maintenance, button to overexploit. Controls growth. |
| Diplomacy | Includes trade and tribute, race status. Controls war/peace, trade, tech collaboration. Sit-rep |
| Influence | Includes all influence projects, race influence status. |
| Shipyards | Includes shipyard construction, upgrades, projects |
| Task forces | Includes task force maintenance, academy training |
| Technology | Research screen |
|  |  |
|  |  |

* Technology
* Influence
* Task force construction
* Infrastructure construction
* Infrastructure maintenance

### Localized developments

* Star
  + Overexploitation
    - Many options by severity (mild, severe) and time (5 turns, 10 turns, 20 turns, forever)
  + Espionage (Galaxy map reveal)
* Movement
  + Hyperspace Gate: allows small ships with no gate to cross to hyperspace.
  + Hyperspace Accelerators: increases speed of friendly outbound ships
  + Hyperspace Dampeners: decreases speed of inbound enemy ships
  + Hyperspace Stabilizers: Minimal engine to survive in hyperspace has nothing to do with speed.

### Global

* Upgrade Ship Design

## Races

### Modifiers

Race modifiers should not be only about bonuses or penalties, they should hopefully be about different ways to play the game. Some ideas:

#### Canibalization

The race consumes worlds. Constantly burn a system’s resources in exchange for a production bonus.

#### Naturists

A planet only grows at his own pace (with local resources).

* No maintenance?
* Scavengers: learn tech from others after any combat.

#### Morale

Bonus or penalty according to the abundance of space and good conditions. Good for new planets but limits the output of older outposts.

#### Corruption

Production penalty based on the size of the empire. This is interesting since it can put larger empires in trouble, especially since there is not much of a balancing act other ways (bigger->better).

## Empire Attributes

|  |  |  |
| --- | --- | --- |
| Attribute | Meaning | Effect |
| Tolerance | Adaptability to hostile or though environments. | Gets added to planet conditions. |
| Research | Capacity to research new ideas and implement them. | Number of research points obtained by a unit of production. |
| Construction | Capacity of building ships, infrastructure and other physical objects. | Number of construction points obtained by a unit of production. |
| Security | Capacity to interest communications, infiltrate other empires, etc. | Number of spying points obtained by a unit of production. |
| Covert Ops | Guerrilla tactics, sabotage, etc. | Decreases the cost of covert operations. |
| Sensors |  | Detection distance from outposts and ships. |
| Speed | Speed at which ships can travel through starlanes. |  |
| Ship Offense |  |  |
| Ship Defense |  |  |
| Ground Offense |  |  |
| Ground Defense |  |  |
| Miniaturization |  |  |
| Training | Ability to train task force officers and |  |
| Morale |  |  |
| Culture |  |  |
| Logistics |  |  |
| Bombardment |  |  |
| Assimilation |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* Also on Moo II
  + Money
  + Food
  + Morale
  + Population
  + Trade
  + Luck
  + Troop experience
  + Command Points

### Fleet design

#### Simple Version

In the simple combat version only few standard ships exist, each equipped with the latest technology and auto-upgrading.

#### Complex Version

Most games allow some kind of ship design. In general the player tends to be restricted to a few ship sizes or “hulls”, that can be filled either to a specific limit of free space or, even worse, in a specific number of placeholders of different types. These limitations seem to be oriented to either:

* Create a complex puzzle mini-game where the player has to deal with design complexity.
* Facilitate the creation of cinematic battles knowing where the beams will be going out of a ship.
* Just screw with the player.

Here we take the opposite approach, a player selects what he wants to get into a ship design and the cost and size is calculated from them. The idea is to reduce complexity and avoid frustrations (the new component I just unlocked doesn’t fit) while being also more realistic (I’m the emperor of a galactic empire god dammit, I should be able to get some engineers to get me the design I want). According to the resulting size, the player can choose between sets of pre-made graphical designs, scaled accordingly.

The creation process involves the following steps:

* Selecting ship weapons/components and their quantities.
* Select the desired speed and defenses, which costs actually depend on the estimated size.
* Check the stats for resulting tonnage, crew size and production cost.

It would be easy for players to try to add more and more stuff, so the idea is to update the final stats automatically on each selection and also provide some feedback as to the expected number of turns (in color) that would take to produce the design given the current military spending and the capacity of his existing shipyards. This feedback should allow the player to decide on the best strategy, but also allow him to create the Death Star if he wants to.

## Space Combat

* 1D combat, option to choose behavior for each ship stack in realtime.
* Intelligently separate ships into a limited number of stacks of each type, based on tonnage.

1. As seen on MoO1 [↑](#footnote-ref-1)
2. An alternative was the one with less trust. While easy to implement, but it complicated granularity when choosing trust, created border cases in case of ties and could be easily exploited to choose which fleet should receive the damage and which one could attack safely. [↑](#footnote-ref-2)